VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI – 590 018

Report

on

"Automated Guided Vehicle"

Submitted in partial fulfilment of the requirements for the award of degree

Bachelor of Engineering in Mechatronics Engineering

Md. Perwez Qaisar 1AY18MT024

Under the guidance of

Dr. Naveen Kumar S N

Acharya Institute of Technology DEPARTMENT OF MEHATRONICS ENGINEERING (Affiliated to Visvesvaraya Technological University, Belgaum) Soldevanahalli, Bengaluru – 560 107

2021 -22

INDEX

Abstract

High demands on manufacturers have left their shipping warehouses in havoc. Human error has a negative effect on safety, efficiency, and quality. These expenses are reduced with the introduction of an Automated Guided Vehicle, AGV. A driverless, intelligent forklift uses an optical path to quickly and safely traverse a warehouse. Its capabilities are enhanced by the ability to send and receive tasks through RF data communication.

INTRODUCTION

Advancements in manufacturing technology allow companies to rapidly produce products. This has provoked a trend to reduce bulk inventory in favour of short-term supplies. Although this allows corporations more financial freedom, it requires warehouses to accommodate temporary, selective storage. Improved product handling and speed can be achieved with the implementation of an Automated Guided Vehicle, AGV. In a traditional warehouse, human safety governs the productivity. With the help of intelligent computers, the AGV can safely achieve higher speeds. Precision turning allows it to accurately navigate in tight spaces. The AGV is highly flexible as a result of remote communication. Its ability to communicate with other autonomous vehicles provides a seamless operation. Continuous coordination between vehicles delivers money saving efficiency. The introduction of unmanned vehicles onto a warehouse floor has favorable effects on safety. With the aid of environmental sensors, the AGV can detect objects in its collision path. Automation eliminates vehicle traffic jams and their potential for accidents. For companies building new warehouses, there are many monetary benefits to investing in intelligent machinery. The workforce required to run the warehouse and the additional overhead (e.g., insurance) required to support that overhead will be drastically reduced. Increased product turn-around and faster shipping will result in more satisfied customers. Also, automation reduces the risks of personal injury. So, an automated guided vehicle system (AGVS) is a material handling system that uses independently operated, self-propelled vehicles guided along defined pathways

Types of automated guided vehicles

Automated guided vehicles can be divided into the following categories:

1) Towing vehicles for driverless trains
2) Pallet trucks
3) Unit load carriers

A driverless train consists of a towing vehicle (the AGV) pulling one or more trailers to form a train. It was the first type of

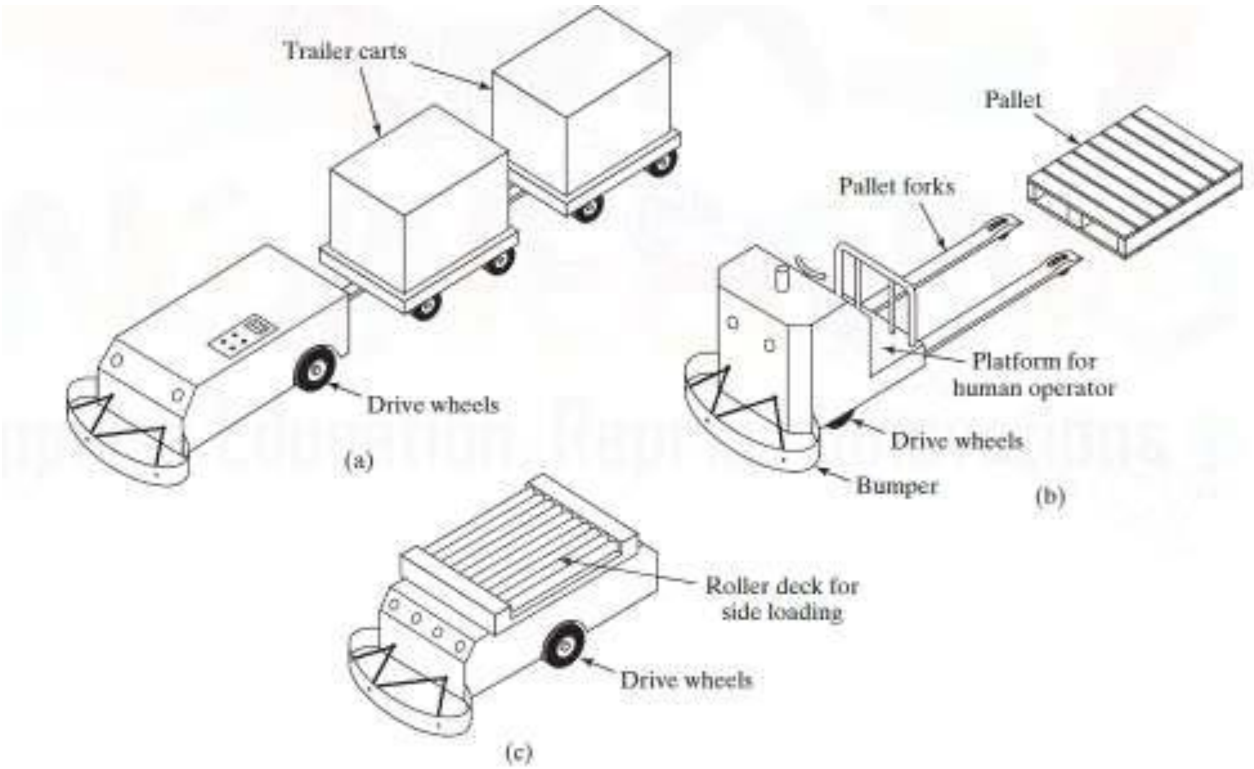
AGVS to be introduced and is still widely used today. A common application is moving heavy payloads over long distances in warehouses or factories with or without intermediate pickup and drop-off points along the route. For trains consisting of 5–10 trailers, this is an efficient transport system.

Automated Guided Pallet trucks are used to move palletized loads along pre- determined routes. In the typical application the vehicle is backed into the loaded pallet by a human worker who steers the truck and uses its forks to elevate the load slightly. Then the worker drives the pallet truck to the guide path and programs its destination, and the vehicle proceeds automatically to the destination for unloading. The load capacity of an AGVS pallet truck ranges up to several thousand kilograms, and some trucks are capable of handling two pallets rather than one.

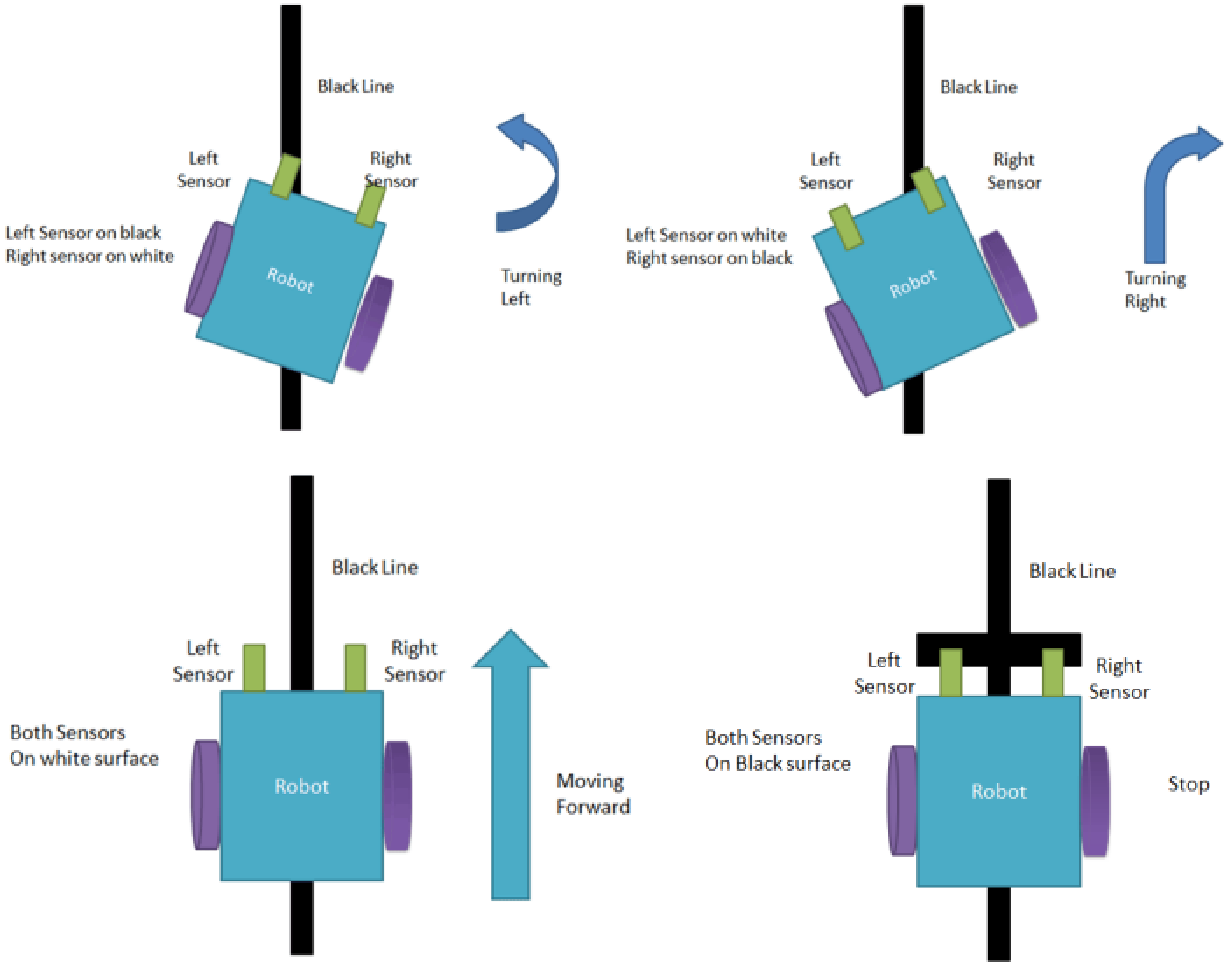
A special type of pallet truck is the forklift AGV, which uses forks that are similar to those of a forklift truck to engage pallets. This vehicle can achieve significant vertical movement of its forks to reach loads on racks and shelves. AGV unit load carriers are used to move unit loads from one station to another. They are often equipped for automatic loading and unloading of pallets or tote pans by means of powered rollers, moving belts, mechanized lift platforms, or other devices built into the vehicle deck.

Variations of unit load carriers include light load AGVs, assembly line AGVs, and heavy-duty AGVs. The light load AGV is a relatively small vehicle with corresponding light load capacity (typically 250 kg or less). It does not require the same large aisle width as a conventional AGV. Light load guided vehicles are designed to move small loads (single parts, small baskets, or tote pans of parts) through plants of limited size engaged in light manufacturing. An assembly line

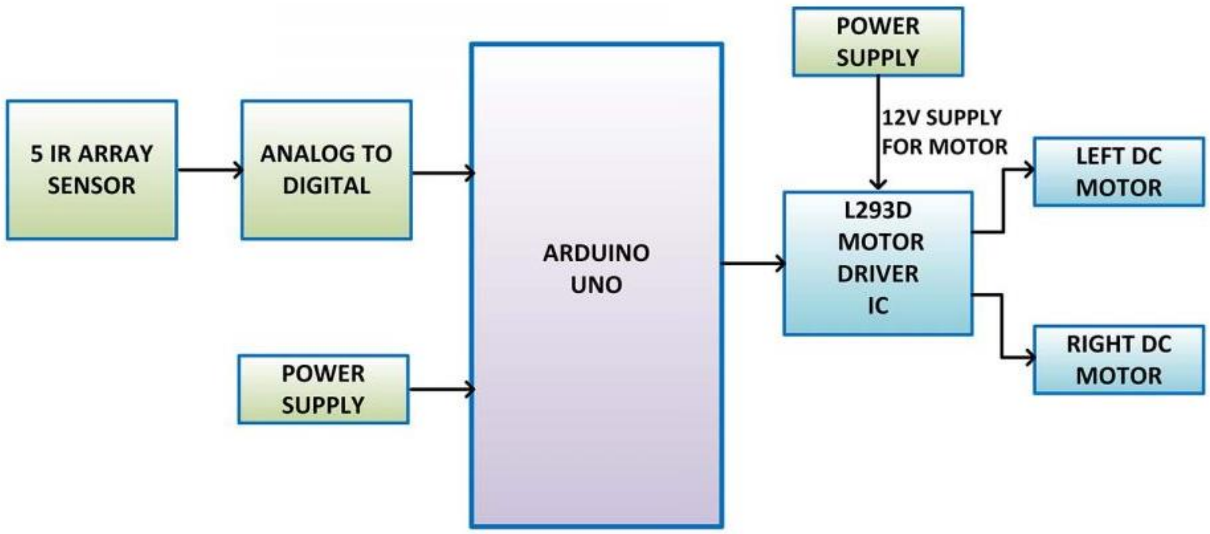
AGV is designed to carry a partially completed subassembly through a sequence of assembly workstations to build the product. Heavy duty unit load AGVs are used for loads up to 125 tons . Applications include moving large paper rolls in printing companies, heavy steel coils in stamping plants, and cargo containers in seaport docking operations. The below figure explains all the types of AGVs.



Working Of Robot



BLOCK DIAGRAM FOR AGV ROBOT



Parts Required

The parts required to build the robot are as follows:

- Chassis (including motors and wheels)
- Arduino Uno r3
- L293D Motor Shield
- IR Proximity Sensors (pair)
- Jumper Wires
- Switch
- 4AA Battery Holder

Tools required:-

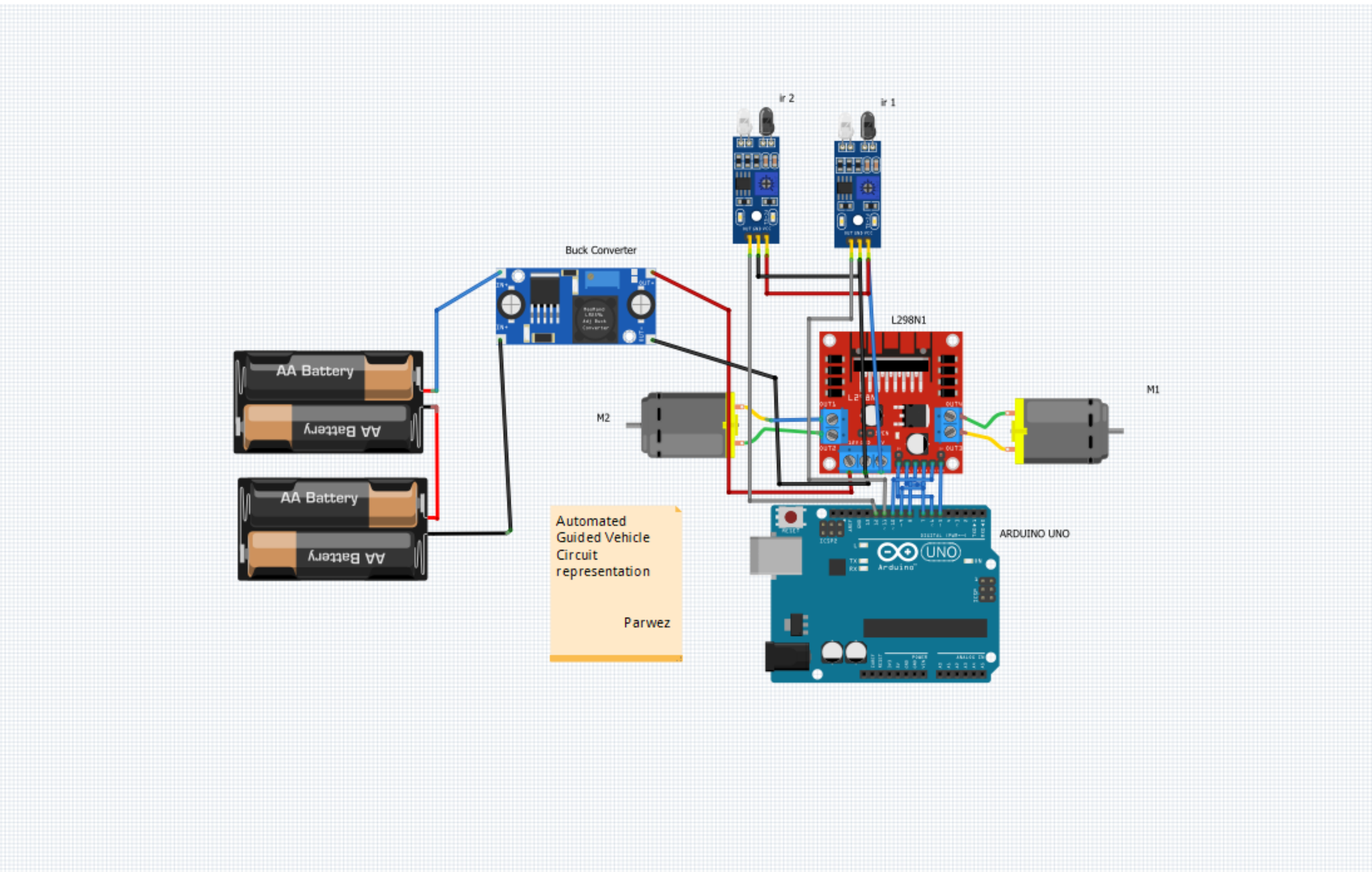
- Soldering Iron
- Hot Glue Gun
- Screw Driver

Assembly

Fig 3 – Chasis Parts To be Assemble

Fig 4 – Assembled Chasis with motors and wheels

Circuit Connection Representation



The connections as per the diagram above

Left Sensor>>Arduino:-

- Vcc>>5v
- Gnd>>Gnd
- Out>>A4

Right Sensor>>Arduino:-

- Vcc>>5v
- Gnd>>Gnd
- Out>>A5

Connect the **motors to the motor shield and plug the motor shield onto the Arduino board**. Connect the **battery holder to the shield through a switch**.

C PROGRAM TO RUN THE ROBOT

```
#define IR_SENSOR_RIGHT 11
#define IR_SENSOR_LEFT 12
#define MOTOR_SPEED 180

//Right motor
int enableRightMotor=6;
int rightMotorPin1=7;
int rightMotorPin2=8;

//Left motor
int enableLeftMotor=5;
int leftMotorPin1=9;
int leftMotorPin2=10;

void setup()
{
  //The problem with TT gear motors is that, at very low pwm value it does not even rotate.
  //If we increase the PWM value then it rotates faster and our robot is not controlled in that speed and goes out of line.
  //For that we need to increase the frequency of analogWrite.
  //Below line is important to change the frequency of PWM signal on pin D5 and D6
  //Because of this, motor runs in controlled manner (lower speed) at high PWM value.
  //This sets frequecny as 7812.5 hz.
  TCCR0B = TCCR0B & B11111000 | B00000010 ;
```

```
  // put your setup code here, to run once:
  pinMode(enableRightMotor, OUTPUT);
  pinMode(rightMotorPin1, OUTPUT);
  pinMode(rightMotorPin2, OUTPUT);

  pinMode(enableLeftMotor, OUTPUT);
  pinMode(leftMotorPin1, OUTPUT);
  pinMode(leftMotorPin2, OUTPUT);

  pinMode(IR_SENSOR_RIGHT, INPUT);
  pinMode(IR_SENSOR_LEFT, INPUT);
  rotateMotor(0,0);
}


void loop()
{

  int rightIRSensorValue = digitalRead(IR_SENSOR_RIGHT);
  int leftIRSensorValue = digitalRead(IR_SENSOR_LEFT);

  //If none of the sensors detects black line, then go straight
  if (rightIRSensorValue == LOW && leftIRSensorValue == LOW)
  {
    rotateMotor(MOTOR_SPEED, MOTOR_SPEED);
  }
  //If right sensor detects black line, then turn right
  else if (rightIRSensorValue == HIGH && leftIRSensorValue == LOW )
  {
      rotateMotor(-MOTOR_SPEED, MOTOR_SPEED);
  }
  //If left sensor detects black line, then turn left
  else if (rightIRSensorValue == LOW && leftIRSensorValue == HIGH )
  {
      rotateMotor(MOTOR_SPEED, -MOTOR_SPEED);
  }
  //If both the sensors detect black line, then stop
  else
  {
    rotateMotor(0, 0);
  }
}


void rotateMotor(int rightMotorSpeed, int leftMotorSpeed)
{
```

```
  if (rightMotorSpeed < 0)
  {
    digitalWrite(rightMotorPin1,LOW);
    digitalWrite(rightMotorPin2,HIGH);
  }
  else if (rightMotorSpeed > 0)
  {
    digitalWrite(rightMotorPin1,HIGH);
    digitalWrite(rightMotorPin2,LOW);
  }
  else
  {
    digitalWrite(rightMotorPin1,LOW);
    digitalWrite(rightMotorPin2,LOW);
  }

  if (leftMotorSpeed < 0)
  {
    digitalWrite(leftMotorPin1,LOW);
    digitalWrite(leftMotorPin2,HIGH);
  }
  else if (leftMotorSpeed > 0)
  {
    digitalWrite(leftMotorPin1,HIGH);
    digitalWrite(leftMotorPin2,LOW);
  }
  else
  {
    digitalWrite(leftMotorPin1,LOW);
    digitalWrite(leftMotorPin2,LOW);
  }
  analogWrite(enableRightMotor, abs(rightMotorSpeed));
  analogWrite(enableLeftMotor, abs(leftMotorSpeed));
}
```